
1	Ziele	6
1.1	Kunde.....	7
1.1.1	Kosten.....	7
1.1.2	Risikominimierung.....	7
1.1.3	Profitabilität.....	8
1.2	Teammitglied.....	10
1.3	Unternehmer	11
	Literatur.....	12
2	Agilität.....	13
2.1	Die agile Idee	14
2.2	Werte, Prinzipien und Praktiken.....	18
2.2.1	Werte.....	18
2.2.2	Prinzipien	18
2.2.3	Praktiken	18
2.3	Begriffseinordnung.....	18
2.3.1	Lean	18
2.3.2	Agilität.....	18
2.3.3	Scrum.....	18
2.3.4	Kanban	18
	Literatur.....	18
3	Scrum Framework und Praxis.....	18
3.1	Die Rollen.....	18
3.1.1	Entwicklungsteam	18
3.1.2	Scrum Master.....	18
3.1.3	Product Owner	18
3.1.4	Nutzer & Stakeholder	18
3.2	Backlog – der Projektumfang.....	18
3.2.1	User Story	18
3.2.2	Job Story.....	18
3.2.3	Story Mapping.....	18
3.2.4	Story Splitting.....	18
3.3	Die Timebox – Der Sprint.....	18
3.4	Produkt-Inkrement.....	18
3.5	Scrum Board.....	18
3.6	Fortschrittsmessung.....	18
3.6.1	Up & Down im Reporting	18
3.6.2	Durchlaufzeit vor Auslastung.....	18
3.7	Rituale.....	18
3.7.1	Refinement	18
3.7.2	Planung.....	18

3.7.3	Daily Standup.....	18
3.7.4	Review	18
3.7.5	Retrospektive	18
3.7.6	Know-how-Transfer.....	18
3.8	Continuous Delivery.....	18
3.9	Schätzen.....	18
3.9.2	Relative vs. Absolute Schätzung	18
3.9.3	Ankerung & Team-Verhalten.....	18
3.9.4	Kilogramm ist der neue Personentag.....	18
3.9.5	Ablauf – Schätzung durchführen Schritt für Schritt	18
3.9.6	Der Druck zum „niedrig Schätzen“	18
3.10	Team-Verträge.....	18
3.10.1	Definition of Ready (DoR)	18
3.10.2	Definition of Done (DoD)	18
	Literatur.....	18
4	Konsequenzen – Unternehmen & Organisation.....	18
4.1	Motivation & gesellschaftliche Veränderungen	18
4.2	Ausrichtung des Unternehmens.....	18
4.3	(Selbst-) Organisation.....	18
4.3.1	Organisation	18
4.3.2	Selbstorganisation.....	18
4.3.3	Soziokratie & Holokratie	18
4.3.4	Transformation vs. Transformierbarkeit.....	18
4.4	Rollenverständnis und Verantwortlichkeiten.....	18
4.4.1	Team	18
4.4.2	Projektmanagement.....	19
4.4.3	Entscheider und Unternehmer	19
4.4.4	Kunde.....	19
4.5	Führung	19
4.6	Entscheidungen	19
4.6.1	Effektivität und Value statt Effizienz und Auslastung	19
4.6.2	Team-Entscheidungen – zentral vs. dezentral	19
4.6.3	Unternehmerische Entscheidungen	19
	Literatur.....	19
5	Skalierung – Mehrere Teams.....	19
5.1	Wir haben Großes vor – mehrere Teams.....	19
5.2	Spezialkenntnisse werden in mehreren Teams gebraucht.....	19
5.3	Skalierungsideen - organisatorisch und methodisch.....	19
5.3.1	SAFe Scaled Agile Framework.....	19
5.3.2	DaD - Disciplined Agile Delivery	19

5.3.3	LeSS – Large Scale Scrum	19
5.3.4	Spotify Fallbeispiel.....	19
5.3.5	EBMgt – Evidence Based Management.....	19
5.3.6	ScALeD – Scaled Agile Lean Development.....	19
5.3.7	Scrum@Scale	19
5.3.8	Nexus	19
5.3.9	XSCALE	19
5.3.10	MAXOS – Matrix of Services.....	19
	Literatur.....	19
6	Agile Kunden-Verträge.....	19
6.1	Kosten-Nutzen Break-Even	19
6.2	Agiler Festpreis & Kalkulation	19
6.3	Maximaler Preis.....	19
6.4	Money for Nothing	19
6.5	Change for Free	19
6.6	Minimal Viable Product (MVP) – Minimalumfang.....	19
	Literatur.....	19
7	Agilität in Nicht-Software-Unternehmen	19
7.1	Produktentwicklung/Hardware	19
7.2	eduScrum	19
	Literatur.....	19
8	Umsetzung.....	19
9	Ausblick – was kommt danach?	20
	Index	20

Aufbau und Inhalt des Buches

Dieses Buch gibt einen Überblick über die agilen Begriffe und ordnet die wesentlichen Begriffe in einen Kontext zueinander.

Ein größerer Block in Kapitel 3 geht auf das Framework Scrum als den bekanntesten Vertreter ein, legt den Schwerpunkt aber auf die Frage des Warum. Ich lege Grundlagen, erkläre zentrale Konzepte und gehe auf die praktische Umsetzung dazu ein.

Um das Konzept Agilität praktisch nutzbar zu machen, gehe ich im Nachgang auf Herausforderungen bei der Umsetzung ein, bespreche die Anwendung, wenn es um mehr als ein einzelnes Team geht und stelle die Skalierungsideen vor.

Wenn die agile Idee die Werte und Prinzipien dann nicht nur in einem Team, sondern an verschiedenen Stellen mehrfach im Unternehmen eingesetzt werden, stellen sich Fragen nach der Integration dieser agilen Teams in die traditionelle Organisation des Unternehmens. Es entstehen Herausforderungen mit der Organisation, den Rollen und dem Führungsstil. Dazu gibt es Ideen und Konzepte, welche ich im Kapitel ► Kap. 0 diskutiere.

Nicht zuletzt gab es im November 2017 ein Update auf dem Scrum Guide. Die Aktualisierung war maßgeblich dadurch getrieben, Scrum als ein handlungsorientiertes Framework auch auf andere Branchen außerhalb der Softwareentwicklung anwenden zu können [2].

Ich greife zwei Beispiele heraus und diskutiere die Anwendung agiler Prinzipien auf die Ausbildung am Beispiel der Schule und der Anwendung bei der Hardwareentwicklung.

Daneben gibt es Community Member, welche diese Prinzipien auf die Baubranche und die Bekämpfung der Korruption adaptiert haben.

Aus Sicht des Nutzers möchte ich darstellen,

- was agiles Mindset ist und wo wirtschaftlicher Nutzen daraus entsteht und
- wie dieses Mindset auf die Organisation einwirkt und wie Sie daraus Nutzen ziehen.

Für den Entscheider lege ich dar, warum der Schwenk auf agile Praktiken unternehmerisch sinnvoll ist. Für Teammitglieder leite ich her, warum ein agiles Mindset ein besseres Arbeitsumfeld schafft.

Ebenso bin ich bemüht, wertvolle Praktiken, Bücher, Konzepte und Personen vorzustellen, die für mich einen wertvollen Betrag zur Verbesserung der Wertschöpfung im Unternehmen geleistet haben.

Das Buch soll einen Überblick über wesentliche Ideen geben, aber auch zum Hinterfragen anregen. Es soll zum kritischen Reflektieren einladen und Gedanken liefern, warum die agilen Ideen, die aktuell so massiv in die unternehmerische Welt drängen, so ein Momentum erzeugen. Was ist da wirklich der Hintergrund?

Ich führe Literatur und Personen aus der Community ein, so können Sie anschließend die für Sie interessanten Konzepte selbst vertiefen.

Teilweise zitiere ich die englischen Originalquellen, wenn ich der Meinung bin, dass diese Formulierungen den Kern besonders gut treffen oder so besonderen Bekanntheitsgrad erlangt haben.

Im Nachgang gehe ich auf die deutsche Übersetzung in der Diskussion ein, oder nenne eine Quelle zur Erläuterung.

Werte

Die agilen Grundwerte wurden im Februar 2001 in einem Ski Resort in Wasatch Mountains in Utah von siebzehn Beteiligten diskutiert. Diese haben anschließend ein Manifest^[1] für agile Softwareentwicklung formuliert.

[1] <http://agilemanifesto.org/>

[2] Ripley, R. (2017, 12 04). <https://ryanripley.com/agile-for-humans/>. Retrieved from Agile for Humans - AFH081 The Scrum Guide gets an UpDate with Dave West: <https://ryanripley.com/afh-081-scrum-guide-gets-update-dave-west/>

“We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.”

In diesem Buch werde ich diese Grundwerte aufgreifen, die maßgeblichen Weiterentwicklungen seit 2001 herausgreifen und die daraus entstandene Idee skizzieren, die alle Industrien weltweit verändert hat und noch weiter verändern wird.

Zu dieser Zeit war der plangetriebene Prozess sehr stark im unternehmerischen Handeln verankert. Diese Statements forderten zum Gegenlenken auf.

Autor

Janko Böhm ist Diplombetriebswirt, hält einen MBA General Business und ist Alumni der German Graduate School of Management and Law. Er war Softwareentwickler im IBM Research Lab, hält ein Patent im Bereich Microcode Entwicklung. Seit 2001 beschäftigt er sich mit Projektmanagement-Methoden, hat die Zertifizierung zum Project Management Professional des PMI abgelegt und beschäftigt sich seit über zehn Jahren mit agilen Methoden in der Theorie und Praxis. Er ist Scrum Master nach Scrum.org und SAFe 4 Certified Scrum Master. Er hat verschiedene mittlere und große Organisationen in seiner Rolle als Interim Geschäftsführer von Unternehmensberatungen geführt, war Gründungsmitglied von Thinking in Projects, sowie Sprecher auf verschiedenen Konferenzen. Aktuell bringt er mit der von ihm gegründeten Organisation methodenfabrik agile Prinzipien in vorrangig deutsche Unternehmen.



1 Ziele

Zusammenfassung

Die zentrale Frage vor der Einführung von agilen Arbeitsweisen sollte eine gute Antwort die Frage nach dem „Warum“ sein. Veränderung bedeutet Anstrengungen. Menschen ändern bekannte und eingeübte Verhaltensweisen nur aufgrund von inneren Überzeugungen. Die Ziele der Menschen in der Organisation müssen mit den Zielen der Organisation übereinstimmen. Nur so tragen Menschen zum Erfolg der Organisation bei und nur so kann ein Unternehmen die richtigen Menschen zur Unterstützung der Ziele der Organisation finden.

Das Kapitel Ziele beschäftigt sich mit der Frage der Motivation zum Einsatz von agilen Prinzipien und Praktiken für die drei Gruppen: Unternehmer als Entscheider in der Organisation, Kunden und Nutzer des erzeugten Produkts und Mitarbeiter als Experte und Teammitglieder in der Herstellung dieses Produkts.

Agilität, Scrum, Kanban sind Begriffe, die sich in der Softwareentwicklung, aber auch in Bereichen der Produkterstellung (Marketing wie auch im Maschinenbau) etabliert haben.

Ich möchte mit diesem Buch die Frage beantworten: „WARUM soll ich das als Unternehmer eigentlich tun?“ Mein Unternehmen arbeitet jeden Tag mit allen Kunden, Projekten und Mitarbeitern in Konkurrenz zu meinen Wettbewerbern.

Als **Entscheider** muss ich dabei auf alle Belange des Unternehmens achten. Qualität, Kosten und Risiko eines Produkts stehen für mich im Vordergrund. Daran werde ich von meinen Kunden gemessen und ich habe auch eine Verantwortung für meine Mitarbeiter.

Mein Kunde formuliert (bisher) die Anforderungen als Ausschreibung, Lastenheft oder Werkleistungsvertrag; auf diesen muss ich mit einer belastbaren Zeit- und Kostenschätzung antworten. In der Einkaufsphase kommt Druck auf Dauer und Kosten-Sätzen des Projektes hinzu.

Wie kann mir in diesem Umfeld die „Agile Idee“ helfen?

Gibt es belastbare Argumente, agile Prinzipien in meiner Produkterstellung einzusetzen?

Entscheider wissen auch um den Faktor „Mitarbeiter“. Bei all den Optionen, die ein Unternehmen hat, werden die verfügbaren Ressourcen die Umsetzung limitieren. Kapital kann vergleichbar recht leicht beschafft werden. Gute Mitarbeiter lassen sich schon schwieriger finden und halten.

Als **Teammitglied** in einem Produktentwicklungsteam möchte ich als Experte wahrgenommen und geachtet werden. Als dieser Experte möchte ich mich auch vorrangig auf meine Aufgaben konzentrieren und nicht durch Prozesse und administrative Aufgaben ausgebremst, bevormundet oder limitiert werden. Ich möchte ein Arbeitsumfeld vorfinden, in dem ich mich selbst auf mein Ziel konzentrieren kann: die Erstellung des Produkts. Das Arbeitsumfeld soll mir die Möglichkeiten geben, mein Wissen und meine Erfahrungen einzubringen. Ich möchte selbst entscheiden wie bestimmte Tätigkeiten ausgeführt werden müssen – denn dazu bin ich hier. Ebenso möchte ich von anderen lernen sowie mein Wissen und meine Fähigkeiten ständig verbessern.

In einer Teamstruktur möchte ich ebenso bei neuen Aufgaben Unterstützung finden und mich so in neue Themen einarbeiten können. Der Austausch mit anderen ist mir wichtig. Ich möchte nicht nur nach Vorgaben und Regeln einer Arbeitsanweisung oder Stellenbeschreibung arbeiten.

Als **Kunde** ist mir wichtig, ein qualitativ hochwertiges Produkt zu wirtschaftlich sinnvollen Kosten zu erhalten. Termintreue ist mir wichtig. Ich erwerbe als B2B-Kunde ein Produkt nicht für mich selbst, sondern weil es einen Beitrag in meiner eigenen Wertschöpfungskette darstellt. Das

Produkt soll möglichst genau auf meine Problemstellung passen. Wenn sich mein Herstellungsprozess verändert, weil ich diesen anpasse oder optimiere, soll sich das Produkt so schnell wie möglich an diese veränderten Bedingungen anpassen. Auch wenn das optimale Produkt noch nicht zur Verfügung steht, möchte ich dennoch bereits früh damit arbeiten.

1.1 Kunde

1.1.1 Kosten

Der große Unterschied zwischen klassischem und agilem Vorgehen ist das Optimierungskriterium für die Herstellungskosten des Produkts. In der klassischen Welt wurden alle Eigenschaften festgelegt, alle Herstellungsschritte abgeleitet und das Kostenoptimum zur Umsetzung errechnet. Für einen festen Umfang ist das nach wie vor das effektivste Optimum. Wenn das Ziel also sehr konkret und belastbar steht, dann ist die plangetriebene Produktion der kosteneffizienteste Weg der Herstellung.

Bei Veränderungen in Technologie, Herstellungsverfahren oder auch den Anforderungen über die Zeit der Herstellung kann dieses Kostenoptimum jedoch nicht mehr aufrechterhalten werden.

Agilität setzt nicht auf die Optimierung der Herstellungskosten und damit der Auslastung der Produktionsressourcen bei einer gegebenen Spezifikation, sondern auf die Optimierung des Nutzens. Die Veränderung ist somit die Verschiebung der Effizienz zugunsten der Effektivität. Es steht dabei immer die Frage im Vordergrund: Welcher Teil des Produkts erzeugt unter den aktuell heute bekannten Fakten den größten Nutzen? Das kann ebenso die Vermeidung von Vertragsverletzungskosten sein – es sind aber in der Regel eher Nutzenaspekte, Features oder Eigenschaften des Produkts, die es dem Endkunden ermöglichen, Profit zu erzielen.

Mit dem agilen Produktentwicklungsansatz können Kosten eingespart werden, weil nur noch der Teil des Produkts erstellt wird, welcher den höchsten Wertschöpfungsteil darstellt. All die Teile, die sonst in einem Lastenheft auch noch benannt wurden und die Fertigstellung verlängert haben, werden jetzt nicht oder evtl. später entwickelt.

Zu einem frühen Zeitpunkt steht schon ein in sich wertschöpfender Teil des Produkts zur Verfügung und kann bereits früh Ertrag generieren. Dieser Ertrag zu einer frühen Phase verbessert den Cash-Flow der Produkterstellung. Die Optimierung auf effektivste Mitarbeiter-Auslastung passiert jetzt nur innerhalb eines kleinen Zeitabschnitts. Innerhalb dieser kleinen Zeiteinheit (z. B. eine oder zwei Wochen) liegt die Verantwortung für diese Optimierung aber nicht beim Management, sondern bei den Experten des Produktentwicklungsteams.

Kennen Sie die „cost of delay“ ihres Produkts? Wie viel Kosten entstehen, wenn das Produkt einen Monat später auf den Markt kommt, als im Businessplan angenommen? Durch Fokussierung auf die Auslieferung von wertschöpfenden Anteilen wird das „alles oder nichts“ aufgebrochen.

cost of delay of delay (Verzugskosten) setzen sich aus zwei Komponenten zusammen[8]:

- Entgangener Gewinn der Zeiteinheit des Verzuges sowie
- Kosten der Produktion in dieser Zeit.

1.1.2 Risikominimierung

Im klassischen Projektmanagement gibt es das Wissensgebiet Risikomanagement. Hier werden in regelmäßigen Abständen potentielle Risiken, aber auch Optionen zum Umgang der Risiken gesammelt, bewertet und Aktionen abgeleitet. Risiken werden zu Issues bzw. Problemen, wenn

die Eintrittswahrscheinlichkeit 100% ist – diese also eingetreten sind. Eine mögliche Strategie zum Umgang mit Risiken ist es, Maßnahmen zu ergreifen, welche die Eintrittswahrscheinlichkeit senken. Die Kosten dieser Maßnahmen werden dem potentiell monetären Risiko gegenübergestellt. Wenn diese Maßnahmen als wirtschaftlich sinnvoll erachtet wurden, werden diese dem Projektplan hinzugefügt und eingeplant.

Im agilen Mindset ist der Umgang mit Risiken direkt im Vorgehen etabliert. Die agile Idee stellt Wertschöpfung in den Fokus. In kurzen Abständen wird Wert geschaffen und das Ergebnis sofort mit dem Kunden zusammen bewertet. Das ist ein Feedback-Loop.

Technische Risiken, welche die Erstellung einer speziellen Funktion oder Eigenschaft behindern könnten, fallen so schnell auf. Es gibt weniger die Bewertung von Möglichkeiten als vielmehr das Sichten des lauffähigen, funktionsfähigen Produktes.

Das Risiko wird durch kurze Iterationen ebenso klein geschnitten. Es kann in einer Iteration von zehn Arbeitstagen, maximal der Wert von zehn Arbeitstagen vernichtet werden. Die Bewertung der Auswirkungen erfolgt auf der Basis der greifbaren Ergebnisse und nicht aufgrund von Bewertungen in Dokumenten.

Das Risiko für fehlende Marktakzeptanz kann direkt mit dem Markt nach dem Einsatz von wenig Zeit und geringen Kosten erfolgen.

Für erfolgskritische Features geht diese Bewertung in die Priorisierung der Features ein. Kritische Elemente werden vorgezogen und direkt validiert.

Das Risiko des Scheiterns eines ganzen Projekts mit mehr als 80% der Kosten wird auf diese Weise effektiv schon durch das Vorgehen selbst verringert. Wenn nach fünf Iterationen noch kein funktionsfähiges, werthaltiges Ergebnis vorliegt, das auch praktisch Wert erzeugt, fällt dies auf. Wenn das Ergebnis beim Endkunden erprobt wird, dieser aber negatives Feedback (zu Qualität, Funktion o. Ä.) liefert, kann dies sofort berücksichtigt werden. Es wird nicht noch weitere Zeit in diese, aus Kundensicht falsche, Richtung entwickelt.

Das Kernprinzip dazu heißt „**Fail Fast**“. Der Grund dahinter ist nicht etwa der Misserfolg, als vielmehr das danach stattfindende Lernen. Die Organisation (Team, aber auch das Unternehmen) profitiert, wenn Fehler als willkommener Hinweis auf Lernmöglichkeiten gesehen werden.

Je schneller Fehler erkannt werden, desto weniger Kapazität und damit auch Geld wird dem Risiko des Scheiterns ausgesetzt.

1.1.3 Profitabilität

Agile Vorgehensweisen können bei richtigem Einsatz zu mehr „Lifecycle-Profit“ als bei der Produktentwicklung nach klassischem Vorgehen führen. Unter Lifecycle-Profit ist der Profit zu verstehen, den ein Produkt über seinen gesamten Lebenszyklus erwirtschaftet. Dabei ist nicht nur die maximale Höhe, sondern auch die Zeit ausschlaggebend, in der ein Produkt am Markt Profite erwirtschaftet.

In klassischem Vorgehen wird ein Produkt in allen seinen Eigenschaften beschrieben. Das wird ein Lastenheft. In der Umsetzung wird dieser Umfang in Tätigkeiten gegliedert und zu einem Plan geformt. In einer Phase der „Umsetzung“ wird der Plan umgesetzt und im erfolgreichen Fall steht das Produkt nun zur Verfügung und kann Wertschöpfung, also Umsatz erzeugen. Nach einer gewissen Zeit hat der erzielte Umsatz durch das Produkt die Betriebskosten und die Herstellungskosten erreicht – es wurde der „Break-even-Punkt“ erreicht. Ab diesem Zeitpunkt erwirtschaftet der Einsatz des Produkts unter Abzug der Betriebskosten Profit.

Auswirkung auf den Profit haben also folgende Elemente der Gleichung:

-
- a) Zeitpunkt der Wertschöpfung - Bereitstellungszeitpunkt
Je früher, desto früher entsteht Umsatz, welcher die Herstellkosten finanziert.
 - b) Höhe der Herstellkosten und Zeitpunkt des Anfallens
Geringere Kosten führen bei gleichem Marktpreis zu höherem Profit.
 - c) Ressourceneinsatz zur Herstellung
Wenn Ressourcen früher frei werden, kann ein neues Produkt umgesetzt werden.
Wenn nur die Elemente umgesetzt werden, die realen Kundennutzen erzeugen, sinkt die absolute Höhe gegenüber einer 100%-Umsetzung nach Pflichtenheft.
 - d) Höhe des Marktpreises / Nutzen für den Kunden
Je mehr Feedback vom Markt zu einem späten Zeitpunkt noch auf das Produkt Einfluss nehmen kann, desto besser kann der Nutzen optimiert werden.

Direkte Unterstützung durch agile Vorgehensweisen:

- 1) Frühere Wertschöpfung
Teilfunktionen des Produkts werden bereits früh bereitgestellt. Es kann Wertschöpfung stattfinden. Ebenso können Feedbacks eingehalten werden, die wieder auf die Priorisierung von Features eingehen können.
- 2) Weniger Produktumfang und geringere Herstellungskosten
Produkteigenschaften oder Features werden nach Wertschöpfung sortiert umgesetzt. So werden zuerst diejenigen Funktionen umgesetzt, die den höchsten Kundennutzen darstellen. Somit wird klar, dass je länger das Projekt mit einem stabilen Umfang läuft, Features umgesetzt werden, die zunehmend weniger Nutzen beinhalten. Die Kosten der Iteration bleiben konstant, da das Team konstant bleibt. Das Verhältnis von Nutzen der umgesetzten Features zu Herstellungskosten wird somit geringer. Das Verhältnis wird vor der Umsetzung jeder Timebox mit dem Team-Commitment durch den Product Owner bestimmt und steht mit dem Abschluss im Review fest.
Wenn die Kosten den Nutzen übersteigen, kann (und sollte) das Projekt beendet werden. Dann wird ab diesem Zeitpunkt mit dem gegebenen Team ein Produkt-Feature erstellt, das einen negativen Deckungsbeitrag erwirtschaftet.
Um das Konstrukt wirtschaftlich zu nutzen, ist es hilfreich, diesen Zeitpunkt im Projekt zu ermitteln. Wenn dieser eintritt, sollte der Auftraggeber vertraglich in der Lage sein, das Projekt zu diesem Zeitpunkt zu beenden. Auf mögliche Vertragsarten gehe ich in ► Kap. 0 ein.
- 3) Zufriedenere Mitarbeiter durch Vermeidung von Auslastungsoptimierung

Bei der klassischen Optimierung von Arbeitsabläufen zur optimalen Auslastung von Maschinen oder Personen wird der Puffer (Slack) zwischen den Tätigkeiten reduziert. Bei einer Störung der optimalen Abläufe muss das System die Neuerstellung eines veränderten Plans, als auch die Mehrarbeiten mit den schon optimal ausgelasteten Ressourcen aufholen. Das führt unweigerlich zu Verzug der Fertigstellung, obwohl die Arbeitsbelastung über 100% steigt.
Auf die Auswirkungen dieses Umstandes gehe ich in Kapitel 1.2 „Aus Sicht des Teammitglieds“ näher ein.

Die Alternative der ressourcenzentrierten Planung ist die Priorisierung von Lieferumfängen nach Kundennutzen. Hier verzichtet man bewusst auf das globale

Optimum in der Effizienz zugunsten einer höheren Effektivität. Eine wesentliche Steuerungseinheit ist nun nicht mehr die Auslastung an einer Station oder Arbeitsplatz, sondern die Liegezeit von halbfertigen Produkten und die Größe der Fertigungseinheiten.

Die praktikable Alternative zum „Durchschieben“ von Arbeitsaufträgen nach einem Push-Verfahren in einen Produktionsablauf ist die Pull-Methode mit Einführung eines Limits der Elemente „in Arbeit“ – siehe „WiP Limit“ im Kanban-Modell.

Auf beide Alternativen gehe ich noch genauer ein.

Diese Argumente verschieben sich noch mehr zugunsten eines agilen Vorgehens, wenn wir neben den Laborbedingungen praktische Elemente wie Änderungen am Lastenheft oder Änderung am Arbeitsumfeld bzw. Technologien unterstellen.

So kann nach dem Bekanntwerden einer Veränderung am Markt und bei den Kundenwünschen durch Neu-Sortierung der Wertschöpfungselemente reagiert werden.

1.2 Teammitglied

Das Mitglied des Entwicklungsteams schafft sich die Arbeitsumgebung, in der zusammen mit den anderen Teammitgliedern die besten Leistungen möglich sind. Nach jeder Iteration können und sollen die Erfahrungen in der Retrospective ausgetauscht werden. Hindernisse werden angesprochen und Abläufe verändert.

Wenn in jeder Iteration durchschnittlich 3% Verbesserung für den erreichten Output bei gleichem Arbeitseinsatz erreicht werden kann, ist das Team auf einem guten Weg.

Dem Team soll durch das Umfeld die Möglichkeit gegeben werden, die Arbeitsabläufe, Werkzeuge, Arbeitszeiten und alle anderen Entscheidungen zum Arbeitsablauf und der konkreten Ausgestaltung selbst zu treffen. Dazu gibt es seit einiger Zeit die Erkenntnis, dass dieser Anspruch eine Umgestaltung der Organisation in den betroffenen Unternehmen in Gang setzt. Führungsstrukturen werden in Frage gestellt. Näher werde ich dazu in Kapitel 0 eingehen.

Ein Werkzeug zur Verhandlung von Rollen und Verantwortlichkeiten, das die Idee der Selbstbestimmung gut aufgreift, ist z. B. Delegation Poker. Mit diesem Werkzeug werden die Gewichte zwischen traditionellem „Management“ und dem Team ausgehandelt [5]. Das Prinzip, ähnlich dem Planning Poker zur Bewertung von Aufwänden, spielt mit den unterschiedlichen Einschätzungen der zwei Gruppen: Management und Team (Näheres dazu in Kapitel 0). Mit diesem Werkzeug werden Bedürfnisse der Teammitglieder transparent, welche nicht mit einer vollen Team-Souveränität und damit voller Verantwortung arbeiten wollen. Je weiter die Verantwortungsbereitschaft auseinander liegt, desto früher entstehen die bekannten Zuständigkeiten „die“ und „wir“ bzw. „ihr“.

Ein Konzept, um einen gesunden Verantwortungsübergang zwischen Produktentwicklung und dem Betrieb zu erreichen, heißt „DevOps“. Das Konzept wurde 2008 auf der Konferenz „Agile Toronto“ von Andrew Shafer and Patrick Debois in Ihrem Beitrag „Agile Infrastructure“ [6] vorgestellt. Ziel dieses Ansatzes ist es, die zwei Bereiche näher zusammenzubringen. Das Entwicklerteam soll sich verantwortlich für den Übergang in die Produktion fühlen und Verantwortung für einen reibungslosen Betrieb übernehmen. Schnellere Entwicklung zugunsten einer hohen Fehlerrate im Betrieb schafft nicht den Kundennutzen, den das Unternehmen erreichen möchte. Wenn die Grenzen verschwimmen und das Produktteam für die Entwicklung als auch für den Betrieb zuständig ist, steht das Produkt im Vordergrund – nicht die Phase des Lebenszyklus.

Diese **Selbstbestimmung** der Verantwortungsbereiche über Rollen und die Verlagerung von Zuständigkeiten näher an das „autonome“ Team bringt für viele Mitarbeiter eine Freiheit, die diese schon lange eingefordert haben. Ebenso gibt es aber auch Stimmen von Teammitgliedern, die mit dieser Art der Organisation und Verschiebung der Verantwortung weniger glücklich sind. Die konkrete Ausgestaltung obliegt der Interaktion zwischen dem Team und der Organisation.

Für diejenigen, welche lieber mit klaren Zuständigkeiten arbeiten, bringt diese Art der Verantwortung zusätzlichen Stress. Hier kann das Team helfen und im Planning II (siehe ► Abschn. 0) die Aufgaben klar zuordnen. Bei Abstimmungsbedarf zwischen den Themen wird das im Daily Standup (siehe ► Abschn. 0) unweigerlich angesprochen. Die Sichtbarkeit der Arbeitsergebnisse ist da. Diese unmittelbare Transparenz kann zu Stress führen. In solchen Fällen hilft der Scrum Master, Team Coach oder eine andere Unterstützung dies aufzufangen. Dennoch wird mit der agilen Arbeitsweise die Transparenz des Arbeitsergebnisses sehr zeitnah sichtbar. Spannungen daraus sind nicht ungewöhnlich.

Das Teammitglied konzentriert sich im agilen Arbeitsumfeld auf die Wertschöpfungsorganisation statt auf die Ablauforganisation. Dazu empfehle ich die Bücher „Komplexithoden“ [4] und „Organisation für Komplexität“ [3] von Niels Pfläging. Er definiert dabei die Wertschöpfungsorganisation als die eigentliche Kette von Mitarbeitern in einer Organisation, die für eine Kundenanfrage Wert schöpfen. Diese Kette läuft im Regelfall nicht an den Organisationslinien des Unternehmens als vielmehr quer durch das Unternehmen. Je besser diese Strukturen unterstützt werden, desto besser können reale Kundenwünsche erfüllt werden. Die Organisation orientiert sich am Kunden, der von außen einen Bedarf formuliert. Sonst geben sich Organisationseinheiten Anforderungen selbst und unterstellen dabei, den Kundenbedarf besser zu kennen als die Kunden selbst. Das ist aber natürlich ein vorgeschobenes Argument, was es schlicht einfacher macht Anforderungen zu definieren und zuweilen hilft, die eigenen Interessen in der Organisation durchzusetzen.

Motivation kann nach Daniel Pink nicht aufgebaut werden. Als Unternehmer, der ein Team aufbaut, kann man lediglich gute Rahmenbedingungen schaffen, dass die motivierten Teammitglieder an der Vision weiter Interesse haben und diese mittragen. Gute und erfolgreiche Projekte entstehen laut Agile Manifest um motivierte Mitarbeiter.

In seinem Buch „Drive“ arbeitet D. Pink dabei drei Erfolgskriterien heraus, die ein selbst motivierter Mensch benötigt [3]:

- Autonomie Entscheidungen selbst treffen zu können
- Purpose Einen Grund für die Arbeit, das Projekt und das Produkt zu sehen
- Mastery Das Können und die Fertigkeiten für die Produktentwicklung mitbringt

1.3 Unternehmer

Zentral für ein Unternehmen ist die wirtschaftliche Stabilität, die ein Unternehmen am Markt bestehen lässt, den Stakeholdern eine Verzinsung ihres eingesetzten Kapitals erwirtschaftet und den Mitarbeitern einen guten, zukunftssicheren Arbeitsplatz sichert, in dem diese sich auch wohlfühlen.

Das Unternehmen braucht Gewinn.

Wenn Produkte (auch Software-Produkte) hergestellt werden, ist die alles entscheidende Steuerungsgröße der (Netto-)Gewinn, den das Produkt über seine Lebensspanne für das Unternehmen erwirtschaftet [9]. Diese KPI ist der „Life Cycle Profit“ (LCP) des Produkts.

Vor dem Hintergrund wird ein Unternehmer seine Organisation nicht mit Veränderungen belasten,

Risiken eingehen und Strukturen umstellen, wenn dadurch nicht nachweislich ein höherer Profit, ein nachhaltigeres Geschäftsmodell, geringere Risiken oder andere messbare Kriterien erreicht werden. Wenn sich ein Mitarbeiter wohler fühlt, wird das betriebswirtschaftlich dann sinnvoll, wenn dadurch die Fluktuationsrate sinkt und damit die Kosten der Personalbeschaffung sinken.

Bei Veränderungen spielt hier also die Messbarkeit eine zentrale Rolle. Wenn Ergebnisse vorliegen, gehört aber auch die Offenheit dazu, diese als Fakten zu sehen, zuzulassen und darüber nachzudenken.

Literatur

- [3] Organisation für Komplexität: Wie Arbeit wieder lebendig wird - und Höchstleistung entsteht Gebundene Ausgabe – 15. August 2014
- [4] Komplexithoden: Clevere Wege zur (Wieder)Belebung von Unternehmen und Arbeit in Komplexität Gebundene Ausgabe – 10. August 2015
- [5] Management 3.0, 09.2018, <https://1qjpt15fhlq3xfpm2utibj1-wpengine.netdna-ssl.com/wp-content/uploads/2016/03/Management30-Delegation-Poker-2015-self-print-A4-German.pdf>
- [6] Agile Infrastructure and Operations Presentation, Patrick Debois, 2008, <http://www.jedi.be/blog/2008/10/09/agile-2008-toronto-agile-infrastructure-and-operations-presentation/>
- [7] Drive: The Surprising Truth About What Motivates Us Paperback – April 5, 2011 by Daniel H. Pink
- [8] „Diving for Hidden Treasures: Uncovering the cost of delay in Your Project Portfolio“ von Johanna Rothman and Jutta Eckstein
- [9] Reinertsen, D. G. (2009). The Principles of Product Development Flow. Redondo Beach, CA: Celeritas Publishing.

2 Agilität

Zusammenfassung

Agilität ist ein weiter Begriff, der von vielen Unternehmen, Beratern und Mitarbeitern auch jeweils anders verstanden und interpretiert wird. Ich möchte hier das Verständnis von Agilität schärfen, dass ich diesem Buch zugrunde lege und ebenso den Begriff von Lean und Scrum abgrenzen. Ausgehend vom agilen Manifesto arbeite ich das Verständnis, die Hintergründe und die Motivation der Begründer heraus. Aus dem Zentrum um die Werte und Prinzipien biete ich praktische Verfahren an, um die Transformation zu starten. Ich biete aus vergangenen Produktentwicklungs-Projekten und Transformationsvorhaben mittlerer und größerer Organisationen praktische Ergebnisse an, die Sie beim Start in die Diskussion als ersten Einstieg direkt so verwenden könne.

Die Welt ist in den vergangenen 15 Jahren schneller, vielfältiger, ungeduldiger und digitaler geworden. Daneben hat sich aber auch das Wissen über Technologie und digitale Medien verbreitert.

Wir wollen nicht mehr warten bis ein neues Angebot vollständig erdacht, finanziert, designt und entwickelt ist, um eine neue Handy-App oder die neue Bohrmaschine endlich in die Hände zu bekommen. Der Verzicht auf Qualität ist dabei nicht die Erwartungshaltung der Nutzer – die Möglichkeit auf ein anderes Produkt zu setzen, ist dabei viel zu nah.

Das eingängige Bild zu Scrum, welche die zwei Kreise Iteration und Daily nach dem Sprintbacklog zeigen, hat mittlerweile bestimmt jeder schon einmal gesehen, der sich mit der Frage nach Entwicklungsmethoden beschäftigt hat.

Was ist denn daran so schwierig – und warum muss man dazu denn so viel schreiben? „Das Prinzip wird ja klar – ich beschäftige mich im täglichen Geschäft schließlich mit wesentlich komplexeren Fragestellungen als diesem einfachen Prozessbild“, werden Sie jetzt eventuell denken.


Agilität und im Speziellen Scrum vergleiche ich dabei gern mit Schach. Die Regeln von Schach passen auf eine Power-Point-Seite – das kann doch nicht so schwierig sein. Ist es auch nicht, aber wer wirklich gut Schach spielen können will, muss sich etwas mehr damit beschäftigen, als die Regeln zu lernen.

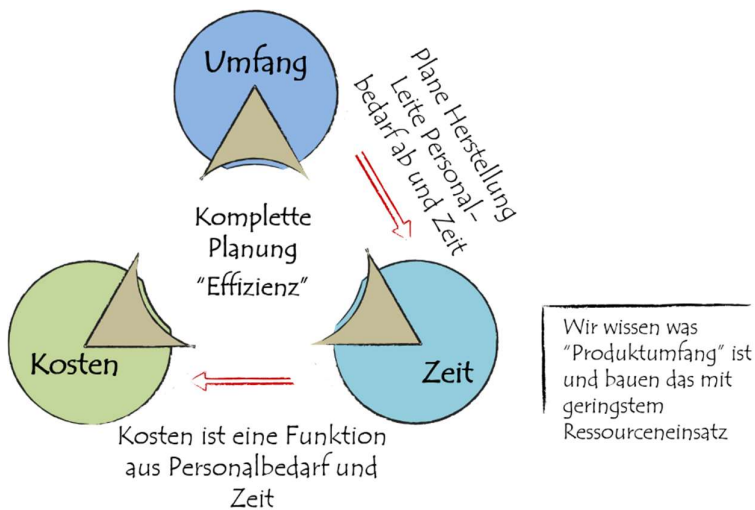
Anstatt nur die Schlagwörter und Begriffe zu können, sollten Sie sich in die Lage versetzen, auf die vielfältigen Situationen in Ihrem Umfeld die Prinzipien anzuwenden. Sie sollen, statt die „Prozesse“ auszuführen, selbst die Prinzipien auf Ihren individuellen Fall anwenden können. Ebenso sollen Sie Ihre Meinung im Team, mit Ihren Kollegen oder sonstigen Beteiligten diskutieren und verteidigen können. Das geht nur, wenn Sie nicht nur die Begriffe, sondern das tiefere „Warum“ verstanden haben.

Das ist dann auch schon der erste der agilen Grundwerte des Agile Manifesto: „Menschen und deren Interaktion über Prozesse und Werkzeuge.“ Prozesse und Werkzeuge sind auch wichtig – wir sehen in Menschen und deren Interaktion aber den größeren Nutzen.

Scrum ist ein Framework aus Prinzipien (Leitsätzen), Artefakten (Bestandteilen) und Ritualen (Verhalten), die es Experten aus unterschiedlichen Gebieten ermöglichen, zur Erreichung des Nutzens effektiv zusammenzuarbeiten.

2.1 Die agile Idee


Das klassische Projektmanagement definiert in einem Lastenheft den Umfang des Projekts, steigt anschließend in die Initiation-Phase ein und erstellt einen Plan für die Umsetzung. Dieser Plan (incl. Ressourcenplan, Kostenplan, Abhängigkeitsplan, Kommunikationsplan usw.) stellt in der Summe den Projektplan dar. Diese Tätigkeit ist aufwändig und kostet einen signifikanten Anteil der Projektlaufzeit als auch am Budget des Projekts. Änderungen am Umfang oder an den Prioritäten der einzelnen Lieferergebnisse führen zwangsläufig zur Überarbeitung dieses Plans und binden erneut Kapazität und Geld. Das Projektmanagement-Dreieck nach  startet beim Umfang und einige Experten planen die Umsetzung komplett durch. Die Planung orientiert sich an Tätigkeiten. Diese Tätigkeiten bekommen Ressourcen und eine Dauer. Aus den Kosten der Ressource und den Angaben zu Dauer und Abhängigkeit werden die Gesamtkosten errechnet. Der Umsetzungsplan baut eine Software „von unten nach oben“. Wenn die zu erstellende Software ein Haus wäre, würde man erst den ganzen Keller, dann das erste Geschoss und dann erst das Obergeschoss und danach das Dach erstellen.



Die Probleme der Herangehensweise sind:

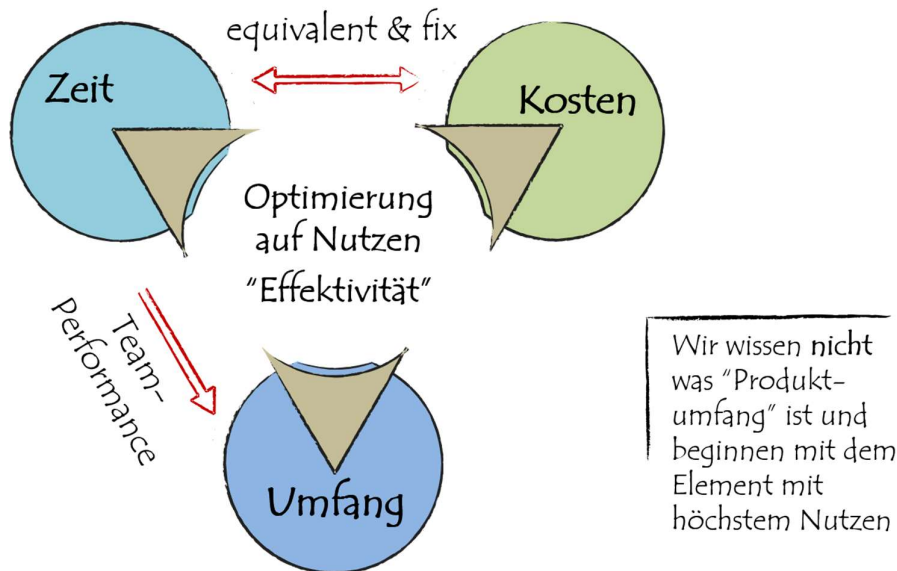
- Zum Zeitpunkt der Planung ist die Performance (Realisierungsdauer) nicht mit Realwerten unterfüttert.
- Der Umfang hinreichend unklar, wurde aber mit „Expertenwissen“ und Annahmen genauer vorgegeben um eine Vollplanung zu ermöglichen.
- Teile des Liefergegenstandes sind nicht nutzbar.
- Alles wird am Ende fertig und nutzbar – hoffentlich.
- Die Entscheidung und damit die Verantwortung über den Umgang mit Veränderungen liegt beim Realisierungspartner, nicht beim Auftraggeber/Kunden.
- Abweichungen von der Planung zeigen sich erst spät.

Das agile Prinzip dreht diese klassische Abfolge auf den Kopf

In  ist die Wertschöpfung der Start – vorgegeben vom Nutzer repräsentiert durch den Auftraggeber, oder noch besser, durch den Kunden. Die Wertschöpfung (Value) ist auch die einzige Messgröße des Projektfortschritts. Es wird nicht die Anzahl von Projekttagen bzw. Dienstleistungsstunden oder Ähnliches genutzt. Maß der Dinge sind einsatzfähige und damit

nutzbringende Funktionen des Lieferergebnisses. Dieses einsatzfähige Lieferergebnis wird dem Kunden übergeben und kann (soll) damit auch sofort Nutzen erzeugen.

Das Framework setzt eine Timebox, in der ein benanntes Team Wertschöpfung erzeugt. Durch die feste Begrenzung dieser Timebox auf z. B. zwei Wochen und dem benannten Team sind Zeit als auch die



Kosten durch die feste Teamgröße exakt definiert. Die Kosten sind auf den Euro vorhersagbar. Der Umfang ist die flexible Seite des Dreiecks. Die Qualität wird nach jeder Iteration durch den Kunden bewertet und ist durch die Team-Norm bzw. Definition of Done festgelegt; dies jedoch nicht durch einen Fragebogen, sondern indem der Kunde das Lieferergebnis einsetzt, um Wertschöpfung zu erzeugen. Die effektivste Kennzahl zur Erfolgsmessung ist damit genau dieser Kundennutzen.

Der Gesamtumfang des Projektes wird auch im agilen Projekt festgehalten; aber die Zeit und der Aufwand, dies ganz besonders detailliert zu tun, entfällt. Es reicht dem Team zu Beginn, den Umfang und damit auch die Komplexität der Herausforderung grob zu verstehen.

Das geflügelte Wort im klassischen Projektmanagement war über lange Zeit, der große Elefant, der in kleine übersichtliche, planbare und ausführbare Scheiben zu schneiden. Die Begrifflichkeit hat 2006 auch die FAZ0 aufgegriffen. Implizit schwingt hier die Annahme mit, die Summe aller Scheiben ergibt dann wieder den großen Elefanten.

Ich verwende gern in Anlehnung an dieses bekannte Bild die Analogie des kleinen Elefanten, der schon nach ganz kurzer Zeit herumlaufen, pusten und kleine Arbeiten verrichten kann. Über die Zeit wird dieser Elefant vom Team gefüttert und wird größer, kann mehr leisten und wird sowohl stärker als auch schneller.


Der Auftraggeber kann nach zehn Scheiben klassischem Elefanten diese noch nicht einsetzen und testen. Bei dem kleinen Elefanten wird eventuell nach 3 Iterationen klar, dass es kein Elefant, sondern eine Giraffe werden soll. Die Erkenntnis kann beim Kunden reifen und die Entscheidung zur Änderung erfolgt nach drei Timeboxes, im Falle des klassischen Projekts evtl. erst am Ende – nachdem alle Scheiben den großen Elefanten montiert sind. Es wurden mehr Zeit und mehr Kosten auf den falschen Elefanten gesetzt.

Das agile Prinzip hinter diesem Bild ist: **„Fail Fast“**

Wenn es in die falsche Richtung geht, soll das möglichst schnell und damit auch mit möglichst

wenig Kosten verbunden sein. Der Rest des Budgets kann dann in ein neues Projekt investiert werden.

Das Gesamtvorhaben (Projekt-Scope) wird in einzelne für sich beschreibbare Einheiten untergliedert. Entgegen der klassischen Entwicklung liegen diese vertikal durch alle Schichten des Softwareprodukts. Beim klassischen Ansatz werden die Schichten nacheinander von unten (Hardware) über die mittleren Schichten (Logik) bis zum kundensichtbaren Layer – der Oberfläche – entwickelt. Das ist effizient.

Nach dem agilen Vorgehensmodell werden aber Elemente entwickelt, welche Nutzen schaffen – diese Einheiten heißen „User Stories“. Jede dieser User Stories kann und wird für die Umsetzung aber Anteile in allen Softwareschichten haben. In  sind die Schichten nach der klassischen Vorgehensweise dargestellt. Auf der rechten Seite der Grafik wird deutlich, dass die einzelnen User Stories vertikal „durch“ alle Software-Layer geschnitten sind. So liefern die einzelnen Elemente Nutzen und das Team entwickelt lediglich den kleinen Teil je Layer, der für genau diesen Teil notwendig ist. Das Vorgehen ist dabei nicht so effizient wie das klassische Vorgehen, weil Dinge in den Layern ggf. umgeschrieben werden müssen, wenn neue Funktionen dazukommen, aber so behält sich das Team zusammen mit dem Kunden vor, die Effektivität zu steigern. Es müssen nicht alle im Lastenheft definierten Features umgesetzt werden. Das Weglassen würde in dem plangetriebenen Vorgehen nicht so ohne weiteres gehen und einen Änderungsprozess starten. Die höhere Effektivität des Vorgehens verursacht in einem änderungsintensiven Umfeld geringere Kosten, weil der Produkt-Scope auf das ausgerichtet werden kann, was den größten Nutzen schafft.

Wenn wir zur Vereinfachung annehmen, dass die Kosten für diese einzelnen User Stories gleich sind, entstehen so die Teile zuerst, welche das beste Kosten-Nutzen-Verhältnis haben. Über die Projektlaufzeit nimmt das Verhältnis ab, da die Elemente, welche einen geringeren Kundennutzen haben, ja später entwickelt werden. Mit Abschluss einer jeden festen Zeiteinheit (Timebox) kann der Kunde erneut entscheiden, ob der Umfang in Summe der Wertschöpfung aus den enthaltenen Liefergegenständen die festen Kosten der Timebox aufwiegt, oder das Projekt den Break Even-Punkt überschritten hat und besser beendet werden sollte.

Mit der agilen Produktentwicklung konzentriert sich das Team zuerst auf das minimal sinnvolle und auslieferbare Produkt (MVP). Die Investitionsphase gegenüber dem klassischen Ansatz wird kürzer gesetzt. Die Investitionssumme (Risiko) ist geringer da die Dauer der Produktentwicklung bis zum ersten Kundenfeedback kürzer ist. So kann weniger Budget (mit gleich großem Team) falsch ausgegeben werden. Nach Auslieferung an den Kunden gibt dieser zur nutzbaren Minimalfunktionalität zeitnah Feedback. Ebenso kann der Kunde diese kleine Funktionalität sofort wertschöpfend einsetzen als auf die (Gesamt-)Funktionalität zu warten die er dann gesamtheitlich aber eben auch erst später einsetzen kann. Dieser Nutzen (in Geld ausgedrückt) läuft der Investition bereits zu diesem frühen Zeitpunkt entgegen. Es steht mehr Zeit zur Verfügung, in der das Produkt bis zum Ende des Lebenszyklus Ertrag erwirtschaftet, weil der Zeitpunkt der Markteinführung vorgezogen wird. Die Rückmeldung des Kunden erfolgt früher. Ein Abbruch nach der Zeit von vier Iterationen hätte im klassischen Modell ein Analysedokument, ein Design-Dokument und die erste Schicht (Hardware/Backend) des Softwareprodukts erzeugt und damit kein nutzbares Kunden-Feature bereitgestellt. Im agilen Modell steht dann ein Produkt mit den Elementen bereit, die den höchsten Kundennutzen darstellen und als minimal sinnvolles Produkt angesehen werden. Eine Feedback-Schleife inklusive Anpassungen am Produkt wird durchgeführt und das Produkt steht zur Nutzung in jeder Iteration für den Kunden in einer bessern Ausbaustufe zur Nutzung bereit.

Wenn der Kunde zu diesem Zeitpunkt feststellt, dass diese Funktionalität bereits ausreicht, kann die Entwicklung jetzt gestoppt werden. Das Budget für die Fertigstellung aller anderen (weniger Nutzen stiftenden) Funktionen wird nicht in dieses Produkt, sondern in ein anderes Produkt investiert.

Am Beispiel des Hauses würde das heißen, das Team erstellt erst ein Zimmer (mit Dach). Das kann man nutzen und dort eine Matratze zum Schlafen aufstellen, Kaffee kochen und wohnen. Weitere Zimmer, die Vergrößerung des Kellers und weitere Zimmer im zweiten Stock kommen später – wenn der Kunde diese dann immer noch möchte. Wenn sich eine Änderung des Plans ergibt und der Kunde doch nur allein mit seinem Hund einziehen möchte – hat das Dachgeschoss noch keine Kosten verursacht. Wenn aber alles genau so, wie vorher geplant entsteht, ist der planungsgetriebene Ansatz der kostengünstigere (effizientere) weil das Dach auf dem ersten Geschoss nicht gebaut worden wäre.

In der Softwareentwicklung heißt dies, jede einzelnen User Story stellt einen Kundennutzen dar. Eine Datenbank zu erstellen, stellt keinen Nutzen dar, wenn nicht auch ein Stück nutzbare Applikation dabei ist. Ob wir im Projekt alle Funktionen umsetzen, für die wir früher das komplette Datenbank-Design erstellt haben, ist unklar. In der Zeit hätten wir ja auch nur die Datenbank-Teile erstellen können, welche für die ersten drei „Killer-Funktionen“ nötig sind und dafür aber im Frontend dann auch nutzbar, statt für 25 Funktionen die Datenbank zu haben – aber keine einzige Funktion vor Kunden nutzen zu können.

Die Konsequenzen

Wenn wir diese Grundidee weiterdenken, ergeben sich einige Konsequenzen, die über die reine Umsetzung von Liefergegenständen weit hinausgehen.

Wegen dieser Konsequenzen wird die agile Idee auch nicht nur von Software-Entwicklern oder Produkt-Entwicklern als Methode der technischen Realisierung diskutiert – die Konsequenzen treffen Personen in bestimmten Rollen wie z.B. Projektleiter, Manager, Teamleiter, Unternehmer und letztlich die ganze Organisation incl. Vertragsgestaltung, Mitarbeiter Incentivierung und viele weitere Themen des Unternehmens.

Projektleiter werden abgeschafft

Das Projektteam entscheidet selbst, wie die Liefsergebnisse umzusetzen sind. Große Projektpläne müssen nicht mehr gerechnet oder aktuell gehalten werden. Die Priorität der Umsetzung orientiert sich am „Was“, den Liefergegenständen, und deren Reihenfolge wird vom Kunden bzw. dessen Repräsentanten festgelegt. Die Aufgabe der Planung verlagert sich. Langfristigere, grobe Planungen erfolgen getrieben vom Markt und den internen Stakeholdern bei einer Rolle, dem Product Owner bzw. eine PO-Organisation. Die kurzfristigeren Planungsaufgaben, die vorrangig die Umsetzung der groben Elemente betreffen, verlagern sich auf das Team.

Das heißt im Umkehrschluss, dass die Fähigkeiten zur langfristigen Planung, Wirtschaftlichkeitsbetrachtung, Methodenwissen sowie der kurzfristigen Umsetzungsplanung weiterhin gebraucht werden, sie konzentrieren sich aber nicht mehr in der einen Management-Person, dem Projektmanager. Die Aufgaben werden anders geschnitten und die Verantwortlichkeiten teilen sich auf.

Teamleiter werden abgeschafft

Das selbstständig arbeitende Team benötigt keine Steuerung durch einen „Team-Manager“, weil es sich in regelmäßigen Abständen trifft und Steuerung anhand der Kundenbedürfnisse stattfindet. Das Team wird ermutigt, eigene Entscheidungen zu treffen und diese auch zu vertreten. Rechte, die notwendig sind, um diese Art der Entscheidungen zu treffen, werden näher an die Stelle verlagert, wo diese Entscheidungen benötigt werden, statt die Varianten und den Kontext an eine andere Stelle in der Organisation zu kommunizieren. Ergebnismessung erfolgt nicht anhand der Arbeit, sondern anhand der Liefergegenständen und dem Kunden-Feedback. Erfolgsmessung kann in Geld erfolgen. Nach jeder Iteration sollte das

potentiell auslieferungsfähige Produkt einen höheren Wert für den Kunden darstellen. Der Betrag, den der Kunde am Markt bereit ist zu zahlen, stellt den Wert des Produkts dar. Der Betrag, den der Kunde nach Abschluss der Iteration bereit ist, für das übergebene Produkt-Inkrement zu zahlen, ist die monetäre Bewertung des Inkrements. Diese sollte über die Zeit die Kosten des Produktteams übersteigen.

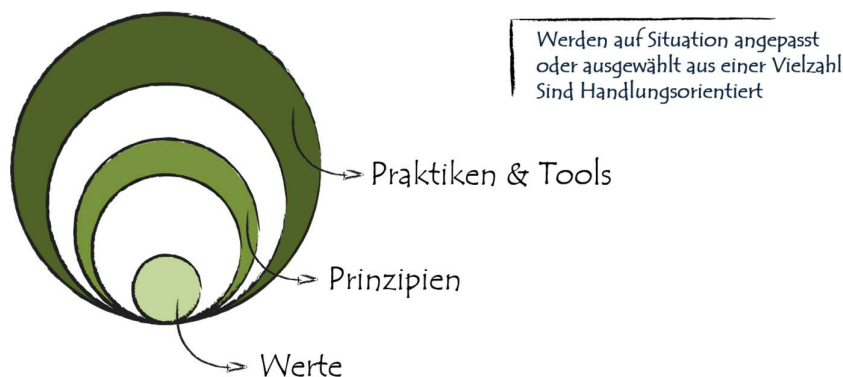
Einige Konsequenzen aus dem Weiterdenken dieser Ansätze sowie einige Lösungsansätze und Anregungen zur Diskussion möchte ich im Kapitel 0 anbieten.

2.2 Werte, Prinzipien und Praktiken

Die Unterscheidung dieser drei sehr wesentlichen Begriffe der agilen Idee ist für das Verständnis wichtig. Während die vier Werte und die zwölf Prinzipien zentrale, fest und nicht handlungsorientierte Grundlagen legen, sind die Praktiken in jedem Projekt, jedem Unternehmen und jedem Team andere. Eine Einordnung dieser Begriffe ist in Abbildung 2 dargestellt und eine Gegenüberstellung der Ziele dieser Konzepte sind in Tabelle 2.1 aufgeführt.

Führung in agilen Unternehmen soll nicht durch Regeln, Prozesse oder Vorschriften erfolgen – sondern werteorientiert. Während lange Dokumente mit Vorschriften eine juristische Berechtigung haben können, werden diese jedoch selten gelesen und noch seltener von denen, die damit erreicht werden sollen. Alle möglichen Situationen vorherzusehen und dafür Regeln und Abläufe zu „planen“, ist schlicht nicht möglich. Mit wenigen Werten aber, z. B. „Wir verhalten uns respektvoll zueinander“, kann mehr transportiert werden als mit fünf Seiten Verhaltensanweisungen.

Werte	Abstract – nicht handlungsleitend (grundlegende Ausrichtung)
Prinzipien	Brücke zwischen Werten und Praktiken, helfen bei der Umsetzung und übersetzen
Praktiken	ganz konkret, handlungsorientiert, variabel, unterliegen Änderungen und werden vom Team ausgewählt und ausgestaltet
[Vergleich von Werten, Prinzipien und Praktiken]	



Werden auf Situation angepasst
oder ausgewählt aus einer Vielzahl
Sind Handlungsorientiert

Bilden den Kern (sind wenige)
Sind über die Zeit stabil
Sind Verhaltensorientiert

Umsetzung

Zusammenfassung

Wertschöpfung entsteht auch mit agilen Prinzipien immer erst dann, wenn diese auch in der Produktentwicklung, in Teams und im Zusammenspiel mit und durch Menschen angewendet werden. Erst mit der Umsetzung entstehen praktische Erfahrung und auch hoffentlich bessere Produkte. In diesem Kapitel möchte ich beschreiben, welche konkreten Schritte mir sinnvoll erscheinen, um die agile Idee in ein Unternehmen und eine Organisation zu tragen. Wo liegen die Hürden und welche Elemente entfalten den größten Nutzen, sind aber dennoch einführbar? In diesem Kapitel – der Umsetzung – möchte ich Ihnen praktisch umsetzbare Tipps geben, sowie aus der Erfahrung heraus Vorschläge für die Auswahl und Reihenfolge der Einführung machen.

Nach der Diskussion des agilen Mindset, des Scrum Frameworks im Speziellen, den Motivationsprinzipien im Team sowie Skalierungsansätzen stellt sich die Frage nach der praktischen Umsetzung.

Wo kann und sollte ich denn zuerst beginnen?
Ist alles gleich wichtig?
Gibt es eine Reihenfolge – was hat sich bewährt?

Nur wer mit der **Umsetzung** beginnt, kann von den Ideen auch lernen und Wert für die tägliche Arbeit daraus ziehen.

Nicht alles auf einmal

Aus der Erfahrung heraus hat sich gezeigt, dass es nicht sinnvoll ist, alles auf einmal umzusetzen, selbst wenn der Support vom Management (denn das gibt es ja zumeist noch) voll da ist. Die verschiedenen Menschen mit ihren unterschiedlichen Standpunkten und Überzeugungen haben unterschiedliche Geschwindigkeiten der Adaption.

Kürzere Projekte

Wenn die bestehende Organisation sehr stark klassisch geprägt ist Projektmanager und Projekt-Management-Offices einsetzt, große Spezifikationen erstellt, lange Planungsphasen hat und auch sehr große Projektbudgets, sollte das nicht alles in Frage gestellt werden.

Agilität ist auch nicht für jedes Vorhaben mit dem Team in jeder Situation die richtige und einzige Herangehensweise. Aus der Erfahrung heraus ist es sinnvoll, zu Beginn die Projektlaufzeiten zu reduzieren, kleinere Projekte zu starten und klassisch durchzuführen. Die Rollen und Werkzeuge bleiben bestehen. Über die Zeit können andere Maßnahmen folgen. Reduzierung der Batch Size führt zu schnellerer Übergabe und Nutzung des kleineren Produkts. So entsteht schneller Feedback. Das Ziel des großen Vorhabens muss fokussiert und ein kleinerer, für sich allein sinnvoller Produktumfang muss definiert werden.

Kleine Arbeitspakete

Warteschlangen können schlecht sichtbar gemacht werden und sind doch für einen Großteil des „Waste“ im Sinne des Lean-Gedankens verantwortlich. Wenn diese nicht sichtbar gemacht werden können, reduzieren Sie die Einheiten, welche von der Organisation bearbeitet werden, und zwar so klein wie möglich und sinnvoll. Wenn Arbeitsaufträge nur noch zwei Tage statt bisher fünf Wochen dauern, ist das großartig. Machen Sie diese kleineren Aufgaben auch sichtbar. Wenn Arbeiten warten müssen, weil diese durch Abhängigkeiten blockiert sind, machen Sie auch das sichtbar. Kleben Sie auf diese Aufgaben an jedem Tag, an dem diese Aufgabe nicht bearbeitet wird, einen Punkt.

Feedback-Zyklen etablieren

Nach abgeschlossenen Einheiten oder einer festgelegten Zeit ist es sinnvoll, ...

Index

Abnahme, 56
Agile Coach, 37
Agilität, 30
Agility Master, 37
Akzeptanzkriterium, 40
Änderungsmanagement, 40
Anreizsystem, 70
Anwendungsfall, 38
Auslastung, 52
Autonomy, 75
batch size, 83
Batch Size, 26
blind spot barrier, 85
Break even Punkt, 9
burn, 49
BurnDown Chart, 25
Cadence, 53
Commitment, 45
Committed, 36
Consent, 76
Continuous Delivery, 56
Cost of delay, 8
Cross-Functional-Team, 74
Cumulative Flow Diagram, 25
Delegation Poker, 10
Deployment, 46
DevOps, 10
Done, 68
Dual Task Interference, 55
Durchlaufzeit, 26, 52
eduScrum, 107
Effektivität, 7
Effizient, 7
ego barrier, 85
Entscheider, 6
Entscheidungen, 83
Entscheidungskompetenz, 36
Epic, 41
Facilitator, 80
Fibonacci, 63
Framework, 53
Führung, 82
Führungskraft, 81
Gherkin, 40
Guild, 59
GWT, 40
Hierarchie, 82
Holokratie, 75
Impediment, 54
information radiator, 50
inspect & adapt, 57
INVEST, 41
involved, 53
Involved, 36
Job Story, 41
Kanban, 32
Kapazitätsschwankungen, 48
Komplexität, 61
Komplexitätsbewertung, 50
Kunde, 7, 81
Kunden-Nutzen, 72
Lastenheft, 39
Lean, 28
Liefertermin, 52
Magic Estimation, 63, 64
Management 3.0, 78
Mastery, 75
Minimal Viable Product, 104

Minimum Viable Product, 104

Mission, 71

Motivation, 11

MVP, 17, 104

No-estimates, 61

Normen, 69

Nutzen, 72

optimale Teamgröße, 36

Out-Tasking, 54

Parallelisierung, 87

Peer-Review, 25

Persona, 39

Planning Poker, 10, 63

Planungssicherheit, 49

Praktiken, 19

Prinzipien, 19

Product Backlog Item, 41

Product Owner, 38

Produkt Inkrement, 45

Projektleiter, 37, 80

Proxy-Kunde, 73

Pull Prinzip, 26

Pull-Prinzip, 81

Purpose, 75

Push-Prinzip, 80

radikale Transparenz, 85

Refinement, 53

Retrospektive, 57

Review, 56

Risiko, 62

Rolle, 39

SAFe, 89

Scope Creep, 104

Scrum, 31

Scrum Board, 46

Selbstbestimmt, 36

Selbstbestimmung, 11

Selbstorganisation, 74

Skalierung, 87

Soziokratie, 76

Sprint Backlog, 45

Stakeholder, 38

Story Map, 42

StoryPoints, 49

Strategie, 71

Task Breakdown, 54

Taxonomie, 41

Teamgröße, 36

Teammitglied, 6

Testgetriebene Entwicklung, 25

The Toyota Production System, 55

Transparenz, 11

UnLearn, 77

Unsicherheit, 62

vanilla Scrum, 35

Velocity, 47, 48

Verlernen, 77

Ver-Lernen, 77

Vision, 71

walking Skeleton, 42

Warteschlangen, 52

Wast, 55

Werte, 19

Wertschöpfung, 113

Wertschöpfungsorganisation, 11

WiP Limit, 25, 47

Working out Loud, 36

Zielkonflikt, 74

Zuständigkeiten, 113

